**PROGRAM NO. 15**

**AIM-** To draw a B-Spline curve.

#include<iostream.h>

#include<conio.h>

#include<graphics.h>

#define max 6

typedefstruct

{

intx,y;}

POINT;

voidknotspline(POINT control[],double knot[],intnum,intklast);

double basis(inti,intk,double knot[],double stpos);

void main()

{

intgm,gd=DETECT;

//control points

POINT control[max]={10,180,80,130,250,10,400,700,500,50,550,70};

//knot vector

double knot[]={0,1,2,3,4,5,6,7};

initgraph(&gd,&gm,"C:\\turboc3\\bgi");

//control,knotvector,no of control points,elements in knot

knotspline(control,knot,max,8);

getch();

closegraph();

}

voidknotspline(POINT control[],double knot[],intnum,intklast)

{

int k=4,i;

doublebas,stpos=knot[k-1],endpos=knot[klast-k],slice=(endpos-stpos)/100;

doublex,y,lx,ly;

lx=control[0].x; //first point

ly=control[0].y;

for(;stpos<endpos;stpos+=slice)

{

x=y=0;

for(i=1;i<=num;i++)

{

bas=basis(i-1,k,knot,stpos);

x=x+(control[i-1].x\*bas); //x,y for bspline curves

y=y+(control[i-1].y\*bas);

}

line(lx,ly,x,y);

lx=x; //last point

ly=y;

}

}

double basis(inti,intk,double knot[],double stpos)

{

doubleval; //recursion

if(k==1)

{

if(knot[i]<=stpos&&stpos<knot[i+1])

return(1);

else

return(0);

}

val=((stpos-knot[i])\*basis(i,k-1,knot,stpos))/(knot[i+k-1]-knot[i])+((knot[i+k]-stpos)\*basis(i+1,k-1,knot,stpos))/(knot[i+k]-knot[i+1]);

return(val);

}

**OUTPUT:-**

